## 1. Introduction

In the extremely competitive game of basketball, The NBA has 30 teams that spend countless millions of dollars on just player salaries alone, yet the spending of all this money doesn't guarantee success. The goal of this project is to investigate the relationship between financial investment (Player Salaries) and on-court performance (Player Stats). Specifically pertaining to the 2022-2023 NBA season. By combining performance statistics with contract data, I aim to determine if higher player salaries correlate with higher individual output or if there are significant discrepancies in this market.

The primary research question guiding this project is: *Does a player's salary accurately predict their scoring contributions? Now to answer this we also had two sub-questions:*

1. *Is there a linear correlation between Annual Salary and Points Per Game (PPG)?*
2. *Which players provided the best "Return on Investment" (ROI) - This is defined as the lowest cost per point scored?*

The approach I used is to utilize Python and the pandas library to integrate two distinct datasets: one of the datasets contain all the players average statistics over the entire regular season and another containing all the players salary information. My analysis consisted of cleaning inconsistent formatting (This includes things like symbols), merging both datasets based on player names, and creating visualizations to identify things like general trends and outliers (undervalued stars).

## 2. Dataset Description

Dataset 1: Player Performance Data

- Name: "2022-2023 NBA Player Stats"
- Source: Kaggle (Originally sourced from Basketball Reference.)
- Content: This dataset contains [679] rows of per game statistics for every player in the NBA. This includes Points (PTS), Assists (AST), and Games Played (G).
- Data Quality & Loading:

Dataset 2: Player Financial Data

- Name: "NBA Salaries: Hoops Fortune (2023-2025)"
- Source: Kaggle
- Content: This dataset contains detailed records of player earnings for each season from 2023-2025. I specifically utilized the data from the 2022/2023 column to match the timeframe of my player performance data.
- Data Quality & Loading: A key obstacle with utilizing this dataset was the file format. While it was labeled as a csv file, the raw data used semicolons (;) as delimiters instead

of commas. As a result loading this dataset correctly required me to use a specific parameter (`sep=';'`) during the loading process to correctly parse the columns.

The 5 V's of the Big Data:

- **Volume:** The combined dataset contains over 460+ players in the NBA, which provides a decently large enough sample size to ensure statistical reliability and minimize the impact of outliers in my findings.
- **Velocity**: Since the data consists of a completed, non-updating record of the 2022-2023 NBA season, it is best analyzed retrospectively.
- **Variety:** This project incorporates two different data types: This includes Performance Metrics (continuous numerical data like Points Per Game (PPG)) and Financial Data (currency values).
- **Veracity:** While the data was generally reliable, there were 'Duplicate' rows for players who were traded mid-season. This means they appeared once for each team. I handled this by filtering to only keep the primary entry for each player to avoid double counting.
- **Value:** Merging these two sources created unique values by allowing us to look at players in a different perspective not just as athletes but as assets essentially. This would reveal the economic efficiency of each NBA team.

## 3. Data Preparation and Exploration Process

**Data Loading and Inspection:**

I used the Python's pandas library to load both datasets. After initially looking through both datasets, I found that there was a formatting issue in the player statistics file. The raw data used semicolons (;) as demiliters instead of the regular commas and this caused the dataframe to load in as a single column. I found this when I ran the 'df.shape()' code and it showed up as (679, 1) which I found strange. I resolved this by specifying in the 'read_csv' function that the separators are the semicolons (;) 'sep=';''. This resolved my issue and the shape of the dataframe was now (679,30).

**Data Cleaning and Manipulation:**

To prepare both datasets for merging, I performed the following data manipulation steps:

- Standardizing Merge Keys: I did this by renaming the 'Player Name' column in the salary dataset to match the 'Player' column in the player statistics dataset. This was a crucial step to do because I was going to the 'Player' columns in both as the key to merge both datasets.
- String Processing: The Salary dataset contained non-numeric characters (For example: "$48,070,014"). I used a regular expression pattern to strip dollar signs and commas, then I converted the column to a floating point number.
- Handling Duplicates: I found that players traded during mid season appeared multiple times in the statistics dataset. To resolve this issue I implemented a filter that would drop

duplicate entries, So the dataset would only retain the primary record for each player to prevent double counting.

**Merging Strategy:**
I performed an Inner Join on the ‘Player’ Column. This means that it would only include the data if the player was found in both datasets. This was a big decision for the validity of the analysis because it ensured that the final dataset included only players with available performance data and salary information. The final dataset contained 467 unique rows, representing the active NBA roster at the time.

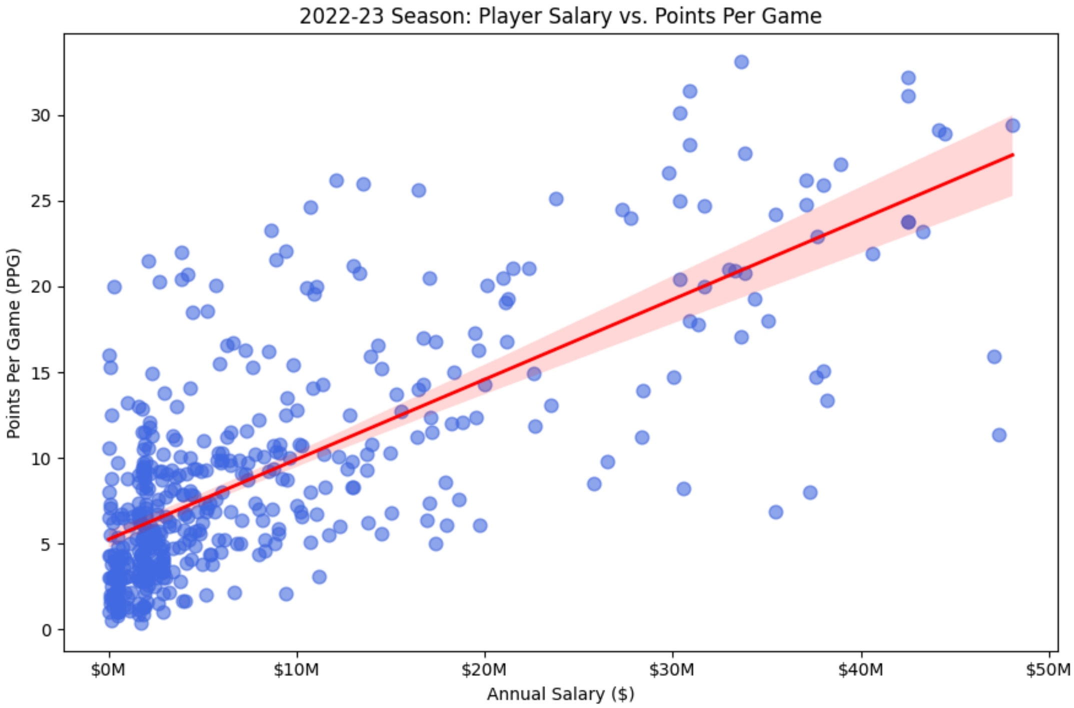
**Feature Construction and Data Preparation:**
To answer the research question regarding the best ROI (Return on Investment), I created a new metric called ‘Cost Per Point’.

- I first had to calculate ‘Total Points’ by multiplying ‘Points Per Game’ (PTS) by ‘Games Played’ (G).
- I then divided ‘Salary’ by ‘Total Points’.
- Handling Edge Cases: To prevent any mathematical errors, I created a duplicate filtered subset and named it ‘df_analysis’. This just specified that I was using this data for the analysis of the project and this subset excluded players that had zero points to prevent any ‘division by zero’ errors throughout the process.

## 4. Analysis and Findings

To address my research question regarding the relationship between financial investment and on-court performance by players, I created three visualizations using the merged dataset.

**Plot 1: Player Salary vs. Points Per Game**

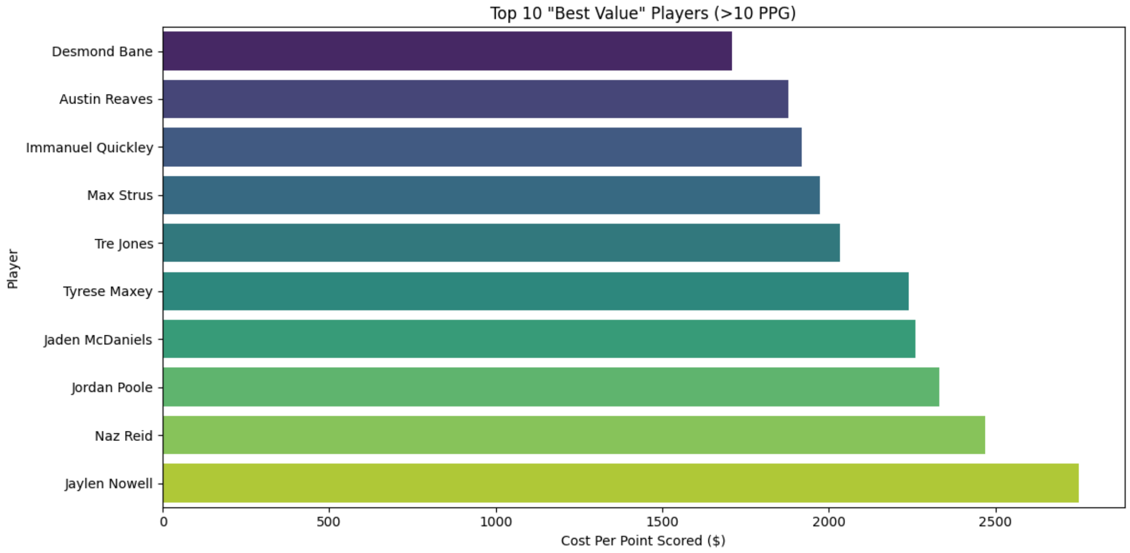


Visualization Choice: I chose to use a scatterplot because it’s one of the most effective ways to show the correlation between two continuous variables. It’s also effective at showing outliers, so you get to see both general trends and outliers. In this case I used it to identify correlations between ‘Salary’ (Financial Investment) and ‘PTS’ (On-Court Performance).

Observation: The plot displayed a generally positive trend which is that as the salary of the players increases, the points per game generally increase as well.

Interpretation: While there is some correlation, it’s not perfectly linear. We see a big cluster of dots in the bottom left of the plot which represents the majority of players in the league and we can see that there is a good amount of variance and outliers meaning players with higher salaries but lower scoring output as well. This would suggest that while money can buy scoring potential it does not guarantee production/performance output.

## Plot 2: Top 10 ‘Best Value’ Players (ROI)

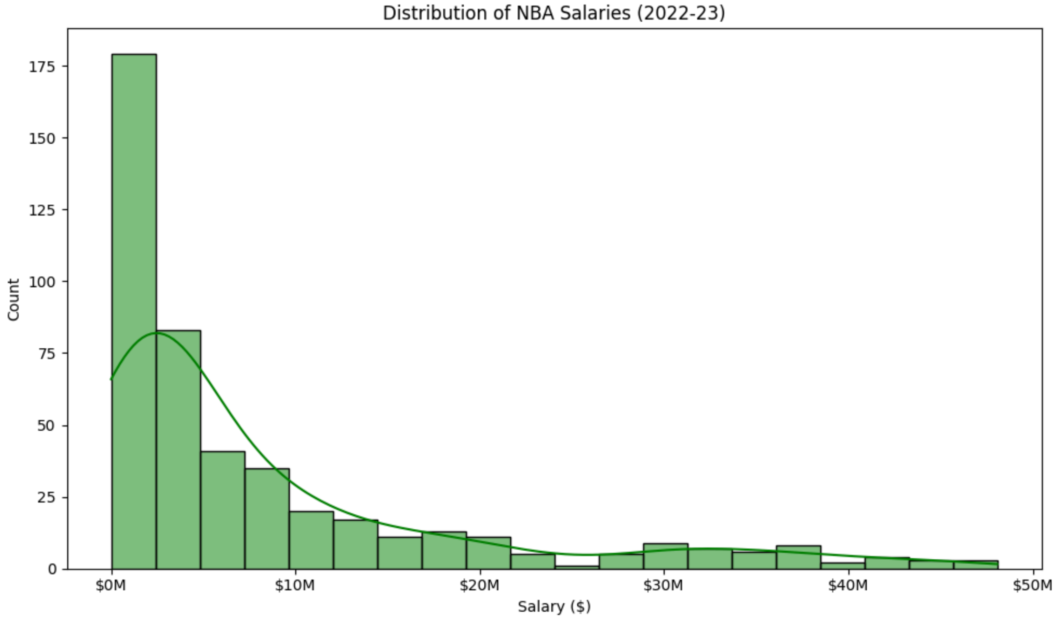


Visualization Choice: I chose a bar chart to represent this because it’s a very effective way to rank players by value and easily highlight and compare the top performers. In this plot we are specifically looking at ‘Player’ and their respective calculated metric which is ‘Cost Per Point’.

Observation: The plot highlights the top 10 players with the lowest cost per point scored (among players scoring > 10 PPG). What this means is it's highlighting the players who have the lowest salaries but produced the best scoring performances.

Interpretation: The results overwhelmingly highlight young players on their initial ‘rookie’ contracts. After doing a little more research the majority of the players on this list are from the 2019 and 2020 draft class. This would mean that they are still on their rookie contract but this would make sense because the NBA caps wages for new players. As a result, these new players going into their second and third year in the league provide the ROI (Return on Investment) for these teams. Which suggests the most efficient way to build a team is through the draft opposed to signing players through free agency.

Plot 3: Distribution of NBA Salaries



Visualization Choice: For this visualization, I chose the histogram because it made the most sense if I wanted to look at the distribution of NBA players salary. This is what the histogram is specifically designed for. This was a crucial visualization for understanding the underlying structure of the data and confirming the 'right-skewed' nature of the NBA payrolls. Oftentimes you only hear about the absurd highest paying contracts.

Observation: By looking at this histogram of NBA salaries we can see the distribution is heavily right-skewed. The tall bars on the left represent the majority of NBA players earning what would be considered minimum wage in terms of the NBA with a long tail extending to the right.

Interpretation: This confirms that the NBA income inequality is extremely drastic. The majority of NBA players earn between $0M and $10M per year, while a tiny percentage of 'Supermax' players earn above $40M per year. This salary structure heavily impacts the team building for these teams because these players take up a lot of the team's salary wage cap limiting flexibility for the rest of the roster.

## 5. Conclusion

**Summary of the Findings:**

My analysis of the 2022-2023 NBA season leads to three main conclusions regarding the economics of player performances:

1. Pay correlates with production: What I mean by this is there is a clear positive relationship between player salary and scoring output. In general, higher paid players tend to produce more points.

2. Market Inefficiencies Exist: This relationship is not perfect. The scatterplot revealed notable outliers, including high paid players producing subpar scoring production. This indicates that large contracts do not consistently guarantee proportional performance.
3. Rookie Contracts Deliver Superior Value: The ‘Cost Per Point’ analysis demonstrated that players on rookie contracts provide the highest ROI (Return On Investment). The most efficient allocation of salary resources is through the successful drafting and player development instead of free agency spending.

**Confidence in Results:**

I have a high degree of confidence in these findings. The final merged dataset contained 467 unique players, representing nearly the entire active NBA roster. Data Validity was strengthened by filtering out players with zero points and resolving duplicate entries caused by mid-season trades. I will acknowledge that there are some limitations to this analysis, the main one being its reliance on points per game as the primary performance metric. There are other factors of player performance that could justify higher salaries including defensive/playmaking statistics and leadership qualities that are vital for a team to succeed.

**Recommendations:**

Based on these results, NBA front offices and GM’s (General Managers) should prioritize the draft capital and player development as core principles in team-building strategies. While supermax contracts are a generally safe way to secure elite talent, the most cost-efficient way to construct a supporting cast is through rookie contracts rather than mid tier veterans, who often provide lower value relative to their cost.

## 6. Reflection on Tools and Process

**How AI tools were used throughout the project:**

Throughout this project, I utilized AI at a level 3 for this course. I leveraged it as a technical accelerator rather than a decision maker, I used it primarily to aid in data cleaning, debugging, workflow efficiency and structure for presenting the data as best as possible. I used AI to refine my initial research question because my original question was focused on a broad relationship between salary and team wins. AI suggested I should analyze ‘Cost Per Point’ at the individual level. I also utilized it for guidance in what would be the best method to clean the ‘Salary’ Column specifically for generating a regular expression that would correctly handle all currency symbols and convert the data to floating-point numbers. When the data failed to load in correctly as in it loaded in with 679 rows and 1 column. I presented that information and it helped me correctly identify and handle the issue which was to add a specific parameter (sep = ‘;’) to the ‘read_csv’ function. Lastly I leveraged AI to help me in creating the visualizations to properly display the information correctly and efficiently.  After completing everything, I had an idea that I had recently just learned about in my data visualization class which is adding a smoother to plots and this would fit perfectly into my scatterplot. I ended up asking AI how I should go about implementing this into my scatterplot.

**What I found helpful versus less useful:**

Obviously the AI was helpful and very effective at syntax generation and diagnosing errors, the main examples being the generation of regular expressions and also resolving the semicolon delimiter issue. AI saved me a ton of time and potential headaches trying to tackle these issues without it. The lecture slides and notes were also useful to touch up on some topics that I needed to refresh my mind on. I did use internet resources to provide better insights and also interpret the data I was working with better. As mentioned in class the AI struggled with contextual consistency. AI finds it hard to handle edge cases and domain specific information. This was apparent when there were a couple instances where the AI provided some code but it automatically generated column names that didn't exist in my environment. This gave me one specific issue of mismatched columns.

**Key Lessons:**

Data Cleaning Importance: Even though I already knew the importance of data cleaning and working with clean data. It's different when you actually have to do the data cleaning yourself and it's not instructions from a class assignment. I had to deal with the semicolon delimiting file and the formatted currency strings taught me the majority of data analysis work happens way before the analysis happens. I've seen this is a common theme across a lot of things, not just data analysis.

I also learned about the importance of having a clean workspace not just in your surroundings but also in the actual project itself. These types of projects make it very easy to make a mess not just with files but also with the code itself. It's crucial to have comments that provide structure to your code. I also found it easier to do this project in google colab instead of something like VScode or Positron. I find the feature in google colab that allows you to separate the blocks of code is very useful and i prefer to have that feature in any future coding project.

**How this project has informed your thinking about future data science work:**

This project has definitely shifted my perspective on future work and honestly the career path that I chose. This project has given me a glimpse into the work that I'll be doing in the future since my interest is in data analytics. I'm going to be doing very similar work in the future to this project so I found this project to be extremely useful in deciding if this was something that I wanted to keep doing as a career.